Binary Representations, Serial Communication and the Atmel 2560
Administration…

• Top Hat or Zyante problems?
Questions?
Quiz
Data Types

• short, long, int: size depends on the particular microprocessor

• In order to be clear about sizes, gcc (our compiler) provides a set of types, including:
  • int8_t 8-bit signed
  • uint16_t 16-bit unsigned

• Use these for our projects!
Atmel Mega2560 Microcontroller
## Atmel Mega2560

### Pinout

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Pinout</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>PORTD (P2)</td>
<td>PIN (P2)</td>
</tr>
<tr>
<td>02</td>
<td>PORTB (P1)</td>
<td>PIN (P1)</td>
</tr>
<tr>
<td>03</td>
<td>PORTA (P0)</td>
<td>PIN (P0)</td>
</tr>
<tr>
<td>04</td>
<td>Port C</td>
<td>PIN (C)</td>
</tr>
<tr>
<td>05</td>
<td>Port D</td>
<td>PIN (D)</td>
</tr>
<tr>
<td>06</td>
<td>Port E</td>
<td>PIN (E)</td>
</tr>
<tr>
<td>07</td>
<td>Port F</td>
<td>PIN (F)</td>
</tr>
</tbody>
</table>

### Pins Usage

- **GND**
- **VCC**
- **A0** - **A15**
- **PORTD** (P2)
- **PORTB** (P1)
- **PORTA** (P0)
- **Port C**
- **Port D**
- **Port E**
- **Port F**

---

**Note:** This table represents the pinout and usage for the Atmel Mega2560 microcontroller. The pins are labeled according to their function and connection to various ports. Please refer to the official documentation for detailed specifications and usage instructions.
Atmel Mega2560

Pins are organized into 8-bit “Ports”:

- A, B, C … L
- But no “I”
Digital Input/Output

• Each port has three special-purpose registers that control its behavior.
• For port B, they are:
  • DDRB: data direction register B
  • PORTB: port output register B
  • PINB: port input B
Data Direction Register: DDRx

- 8-bit wide register
  - Controls one pin with each bit
- 0 -> this is an input pin
- 1 -> this is an output pin
Port Output Register: PORTx

• Also one pin per bit

• If configured as an output:
  • 0 -> the pin is held at 0 V
  • 1 -> the pin is held at +5 V

• Note: only configure pins as an output if you really mean it!
Port INput register: PINx

• One pin per bit
• Reading from the register:
  • 0 -> the voltage of the pin is near 0 V
  • 1 -> the voltage of the pin is near +5 V
• If nothing is connected to the pin, then the pin will appear to be in a random state
A First Program

Flash the LEDs at a regular interval

• How do we do this?
main() {  
DDRC = 0x3;

while(1)  
{
    PORTC = 0x1;
    delay_ms(100);
    PORTC = 0x0;
    delay_ms(100);
}
}
A First Program

```c
main() {
    DDRC = 0x3;

    while(1) {
        PORTC = 0x1;  // sets PC0 to 1
        delay_ms(100);
        PORTC = 0x0;  // set PC0 to 0
        delay_ms(100);
    }
}
```
A First Program

```c
main() {
    DDRC = 1;   // Set port C pin 0 as an output

    while(1) {
        PORTC = PORTC ^ 0x1;   // XOR bit 0 with 1
        delay_ms(500);         // Pause for 500 msec
    }
}
```
A Second Program

```c
main() {
    DDRC = 3; // Set port C pins 0, and 1 as outputs

    while(1) {
        PORTC = 0x3;
        delay_ms(250);
        PORTC = 0x1;
        delay_ms(250);
        PORTC = 0x2;
        delay_ms(250);
        PORTC = 0x0;
        delay_ms(250);
    }
}
```

What does this program do?
A Second Program

main() {
    DDRC = 3;   // Set port C pins 0, and 1 as outputs

    while(1) {
        PORTC = 0x3;
        delay_ms(250);
        PORTC = 0x1;
        delay_ms(250);
        PORTC = 0x2;
        delay_ms(250);
        PORTC = 0x0;
        delay_ms(250);
    }
}

Flashes LED on PC1 at 2 Hz
on PC0: 1 Hz

Duty Cycle for each: 50%
A Third Program

If switch reads zero, turn PC0 on and PC1 off

Otherwise, turn PC0 off and PC1 on
A Third Program

main() {
    DDRC = 0x3;

    while(1){
        if(PINC & 0x80) {
            PORTC = 1;
        }else{
            PORTC = 2;
        }
    }
}
A Third Program

```c
main() {
    DDRC = 0x3;

    while(1) {
        if(PINC & 0x80) {
            PORTC = 0x2;
        } else {
            PORTC = 0x1;
        }
    }
}
```
Port-Related Registers

Some of the C-accessible registers for controlling digital I/O:

<table>
<thead>
<tr>
<th></th>
<th>Directional control</th>
<th>Writing</th>
<th>Reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port B</td>
<td>DDRB</td>
<td>PORTB</td>
<td>PINB</td>
</tr>
<tr>
<td>Port C</td>
<td>DDRC</td>
<td>PORTC</td>
<td>PINC</td>
</tr>
<tr>
<td>Port D</td>
<td>DDRD</td>
<td>PORTD</td>
<td>PIND</td>
</tr>
</tbody>
</table>
Arduino Mega Board

(see schematic)